

Claims

What is claimed is:

1. A method of switching queue ownership, said method comprising:

obtaining an indication that a queue is to be taken over, said queue being resident in memory of a first processor; and

moving said queue from said first processor to a second processor, said queue to be resident in memory of said second processor.

2. The method of claim 1, wherein said first processor is inactive.

3. The method of claim 1, further comprising rebuilding said queue prior to moving said queue.

4. The method of claim 3, wherein said rebuilding comprises using a recovery log and a checkpoint of said queue to rebuild contents of said queue, said recovery log and said checkpoint being associated with said first processor.

5. The method of claim 3, wherein said rebuilding comprises:

processing at least a portion of a recovery log of said first processor to obtain one or more in-doubt events for the queue; and

merging at least one in-doubt event of the one or more in-doubt events with a checkpoint of the queue to obtain a rebuilt version of the queue, said checkpoint being associated with said first processor.

6. The method of claim 5, further comprising writing the rebuilt version of the queue to the checkpoint of the queue.

7. The method of claim 5, wherein said processing comprises processing said at least a portion of the recovery log in reverse order.

8. The method of claim 5, wherein each in-doubt event of the one or more in-doubt events represents a most recent event for a message of the queue.

9. The method of claim 1, wherein said moving comprises reading the queue into memory of said second processor.

Abstract

determining whether a queue having the defined name already exists on the second processor; and

11. The method of claim 10, wherein said moving contents to the existent queue comprises moving the contents within one or more commit scopes.

writing a version of the queue to a
checkpoint of the second processor; and

POU920000197US1

13. The method of claim 1, further comprising:

writing a version of the queue to a
checkpoint of the second processor; and

deleting a version of the queue from a
checkpoint of the first processor.

POU920000197US1

14. A method of reconstructing queues, said method comprising:

rebuilding contents of a queue to obtain an updated version of the queue, said queue being a memory resident queue of a first processor; and

reading at least a portion of the updated version of the queue into memory of a second processor, said second processor being different than said first processor.

15. The method of claim 14, further comprising writing a version of the queue stored in memory of said second processor to a checkpoint of the second processor.

16. The method of claim 15, further comprising deleting a version of the queue from a checkpoint of the first processor.

17. The method of claim 16, wherein the version of the queue being deleted is an updated version written to the checkpoint of the first processor, in response to the rebuilding.

18. The method of claim 14, wherein said rebuilding comprises:

processing at least a portion of a recovery log of said first processor to obtain one or more in-doubt events for the queue; and

merging at least one in-doubt event of the one or more in-doubt events with a checkpoint of the queue to obtain the updated version of the queue, said checkpoint being associated with said first processor.

19. The method of claim 18, wherein said processing comprises processing said at least a portion of the recovery log in reverse order.

20. The method of claim 14, wherein said rebuilding comprises using a recovery log and a checkpoint of said queue to rebuild contents of said queue, said recovery log and said checkpoint being associated with said first processor.

21. A system of switching queue ownership, said system comprising:

means for obtaining an indication that a queue is to be taken over, said queue being resident in memory of a first processor; and

means for moving said queue from said first processor to a second processor, said queue to be resident in memory of said second processor.

22. The system of claim 21, wherein said first processor is inactive.

23. The system of claim 21, further comprising means for rebuilding said queue prior to moving said queue.

24. The system of claim 23, wherein said means for rebuilding comprises means for using a recovery log and a checkpoint of said queue to rebuild contents of said queue, said recovery log and said checkpoint being associated with said first processor.

25. The system of claim 23, wherein said rebuilding comprises:

means for processing at least a portion of a recovery log of said first processor to obtain one or more in-doubt events for the queue; and

means for merging at least one in-doubt event of the one or more in-doubt events with a checkpoint of the queue to obtain a rebuilt version of the queue, said checkpoint being associated with said first processor.

26. The system of claim 25, further comprising means for writing the rebuilt version of the queue to the checkpoint of the queue.

27. The system of claim 25, wherein said means for processing comprises means for processing said at least a portion of the recovery log in reverse order.

28. The system of claim 25, wherein each in-doubt event of the one or more in-doubt events represents a most recent event for a message of the queue.

29. The system of claim 21, wherein said means for moving comprises means for reading the queue into memory of said second processor.

30. The system of claim 29, wherein said queue has a defined name, and wherein said means for reading comprises:

means for assigning a temporary name to the queue;

means for determining whether a queue having the defined name already exists on the second processor; and

means for moving, in response to the determining, contents of said queue having the assigned temporary name to the existent queue.

31. The system of claim 30, wherein said means for moving contents to the existent queue comprises means for moving the contents within one or more commit scopes.

32. The system of claim 30, further comprising:

means for writing a version of the queue to a checkpoint of the second processor; and

means for deleting a version of the queue from a checkpoint of the first processor.

33. The system of claim 21, further comprising:

means for writing a version of the queue to a
checkpoint of the second processor; and

means for deleting a version of the queue
from a checkpoint of the first processor.

POU920000197US1

34. A system of reconstructing queues, said system comprising:

means for rebuilding contents of a queue to obtain an updated version of the queue, said queue being a memory resident queue of a first processor; and

means for reading at least a portion of the updated version of the queue into memory of a second processor, said second processor being different than said first processor.

35. The system of claim 34, further comprising means for writing a version of the queue stored in memory of said second processor to a checkpoint of the second processor.

36. The system of claim 35, further comprising means for deleting a version of the queue from a checkpoint of the first processor.

37. The system of claim 36, wherein the version of the queue being deleted is an updated version written to the checkpoint of the first processor, in response to the rebuilding.

38. The system of claim 34, wherein said means for rebuilding comprises:

means for processing at least a portion of a recovery log of said first processor to obtain one or more in-doubt events for the queue; and

means for merging at least one in-doubt event of the one or more in-doubt events with a checkpoint of the queue to obtain the updated version of the queue, said checkpoint being associated with said first processor.

39. The system of claim 38, wherein said means for processing comprises means for processing said at least a portion of the recovery log in reverse order.

40. The system of claim 34, wherein said means for rebuilding comprises means for using a recovery log and a checkpoint of said queue to rebuild contents of said queue, said recovery log and said checkpoint being associated with said first processor.

41. A system of switching queue ownership, said system comprising:

a queue resident in memory of a first processor; and

at least a second processor adapted to move said queue from said first processor to the second processor, said queue to be resident in memory of said second processor.

POU920000197US1

42. A system of reconstructing queues, said system comprising:

a first processor adapted to rebuild contents of a queue to obtain an updated version of the queue, said queue being a memory resident queue of a second processor; and

a third processor adapted to read at least a portion of the updated version of the queue into memory of the third processor, said third processor being different than said second processor.

43. The system of claim 42, wherein said first processor and said third processor are the same.

44. The system of claim 42, wherein said first processor and said third processor are different processors.

45. At least one program storage device readable by a machine, tangibly embodying at least one program of instructions executable by the machine to perform a method of switching queue ownership, said method comprising:

obtaining an indication that a queue is to be taken over, said queue being resident in memory of a first processor; and

moving said queue from said first processor to a second processor, said queue to be resident in memory of said second processor.

46. The at least one program storage device of claim 45, wherein said first processor is inactive.

47. The at least one program storage device of claim 45, wherein said method further comprises rebuilding said queue prior to moving said queue.

48. The at least one program storage device of claim 47, wherein said rebuilding comprises using a recovery log and a checkpoint of said queue to rebuild contents of said queue, said recovery log and said checkpoint being associated with said first processor.

49. The at least one program storage device of claim 47, wherein said rebuilding comprises:

processing at least a portion of a recovery log of said first processor to obtain one or more in-doubt events for the queue; and

merging at least one in-doubt event of the one or more in-doubt events with a checkpoint of the queue to obtain a rebuilt version of the queue, said checkpoint being associated with said first processor.

50. The at least one program storage device of claim 49, wherein said method further comprises writing the rebuilt version of the queue to the checkpoint of the queue.

51. The at least one program storage device of claim 49, wherein said processing comprises processing said at least a portion of the recovery log in reverse order.

52. The at least one program storage device of claim 49, wherein each in-doubt event of the one or more in-doubt events represents a most recent event for a message of the queue.

53. The at least one program storage device of claim 45, wherein said moving comprises reading the queue into memory of said second processor.

54. The at least one program storage device of claim 53, wherein said queue has a defined name, and wherein said reading comprises:

assigning a temporary name to the queue;

determining whether a queue having the defined name already exists on the second processor; and

moving, in response to the determining, contents of said queue having the assigned temporary name to the existent queue.

55. The at least one program storage device of claim 54, wherein said moving contents to the existent queue comprises moving the contents within one or more commit scopes.

56. The at least one program storage device of claim 54, wherein said method further comprises:

writing a version of the queue to a checkpoint of the second processor; and

deleting a version of the queue from a checkpoint of the first processor.

57. The at least one program storage device of claim 45, wherein said method further comprises:

writing a version of the queue to a
checkpoint of the second processor; and

deleting a version of the queue from a
checkpoint of the first processor.

POU920000197US1

58. At least one program storage device readable by a machine, tangibly embodying at least one program of instructions executable by the machine to perform a method of reconstructing queues, said method comprising:

rebuilding contents of a queue to obtain an updated version of the queue, said queue being a memory resident queue of a first processor; and

reading at least a portion of the updated version of the queue into memory of a second processor, said second processor being different than said first processor.

59. The at least one program storage device of claim 58, wherein said method further comprises writing a version of the queue stored in memory of said second processor to a checkpoint of the second processor.

60. The at least one program storage device of claim 58, wherein said method further comprises deleting a version of the queue from a checkpoint of the first processor.

61. The at least one program storage device of claim 60, wherein the version of the queue being deleted is an updated version written to the checkpoint of the first processor, in response to the rebuilding.

62. The at least one program storage device of claim 58, wherein said rebuilding comprises:

processing at least a portion of a recovery log of said first processor to obtain one or more in-doubt events for the queue; and

merging at least one in-doubt event of the one or more in-doubt events with a checkpoint of the queue to obtain the updated version of the queue, said checkpoint being associated with said first processor.

63. The at least one program storage device of claim 62, wherein said processing comprises processing said at least a portion of the recovery log in reverse order.

64. The at least one program storage device of claim 58, wherein said rebuilding comprises using a recovery log and a checkpoint of said queue to rebuild contents of said queue, said recovery log and said checkpoint being associated with said first processor.

* * * * *